Part 3

Introduction to statistical classification techniques

Preamble

Note: in statistics $\boldsymbol{D}$ is often called the "sample" of size n drawn from the distribution p(x). In pattern recognition the term "sample" is usually used for the single pattern x_i.

- In Part 2 we have seen that if we know:
  - Posterior probabilities $P(\omega_i / \boldsymbol{x})$
  - Or the equivalent terms $P(\omega_i)\, p(\boldsymbol{x} / \omega_i)$
  - And we know the loss matrix Λ

- The minimum risk theory allows us to design the optimal classifier (that minimizes the classification risk) for the task at hand
- However, in practical cases, we never know all this information
- The only information that we usually have is a data set $\boldsymbol{D}$ (called "design" or "training" data set)

$$\boldsymbol{D} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$$

$$\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{id}) \quad i=1,..,n$$

$\boldsymbol{x}_i$ belonging to one of the "c" classes ($\boldsymbol{x}_i \in \omega_j \quad j=1,\ldots,c$)

- Patterns $\boldsymbol{x}_i$ are drawn "independently" according to $p(\boldsymbol{x}_i / \omega_j)$

Classification techniques

➢If we know the "classes" to which the patterns $\boldsymbol{x}_i$ of the design/training set belong to, we speak of **Supervised Classification**

Further information, beyond the data sert $\boldsymbol{D}$, that we can have are:

- ➢ We can know the parametric model ("parametric form") of the distribution $p(\boldsymbol{x}/\omega_i)$, so that we can use the **Parametric Techniques**
- ➢ If we know nothing about the distribution $p(\boldsymbol{x}/\omega_i)$, we are obliged to use the so called **Non-Parametric Techniques**

➢**Parametric Techniques**: we know the parametric form of the distribution $p(\boldsymbol{x}/\omega_i)$, for example, we know that the distribution is Gaussian

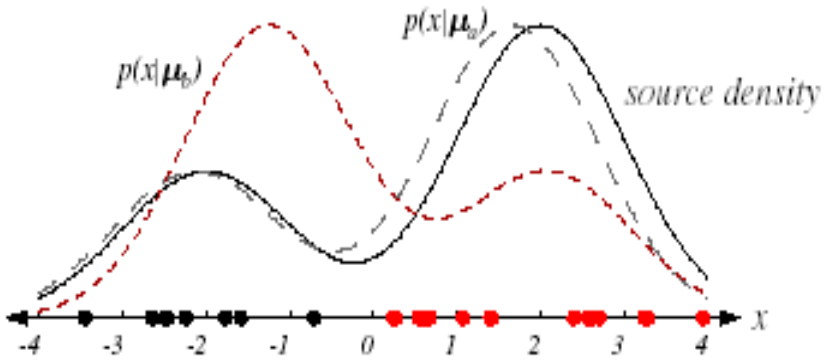
➢**Non-Parametric Techniques**: we know nothing about the distribution, and we are not able to get any information with an unsupervised analysis.

•Note that we are assuming that estimating priors $P(\omega_i)$ is an easy problem, assumption that is often but not always true.

➢Here we are disregarding the "costs" of classification. The reason is that the choice of "cost" values is a problem-dependent issue, very little can be said in general about this choice.

Classification Parametric Techniques

- We know, or we assume, a parametric form of the distributions $p(\boldsymbol{x}/\omega_i)$.
- The main problem is then to estimate the **parameter**s of the model (e.g., the mean value and the variance of the Guassian model)
- We discuss in detail these techniques in Part 4.
- The estimate of the parameters is done using the data set $\boldsymbol{D}$, or a subset of it more often (to avoid a problem called "over-fitting").
- How can we assume a good parametric model of the distributions $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$? In the practical applications we have two possibilities to do that:
  - We assume different parametric models, we compute the parameters for each model, then we compare the errors of the models and select the best
  - We use **Unsupervised Classification Techniques** (we see basic concepts later in Part 9) to gain some knowledge of the parametric form of the $p(\boldsymbol{x}/\ \boldsymbol{\omega}_i)$.



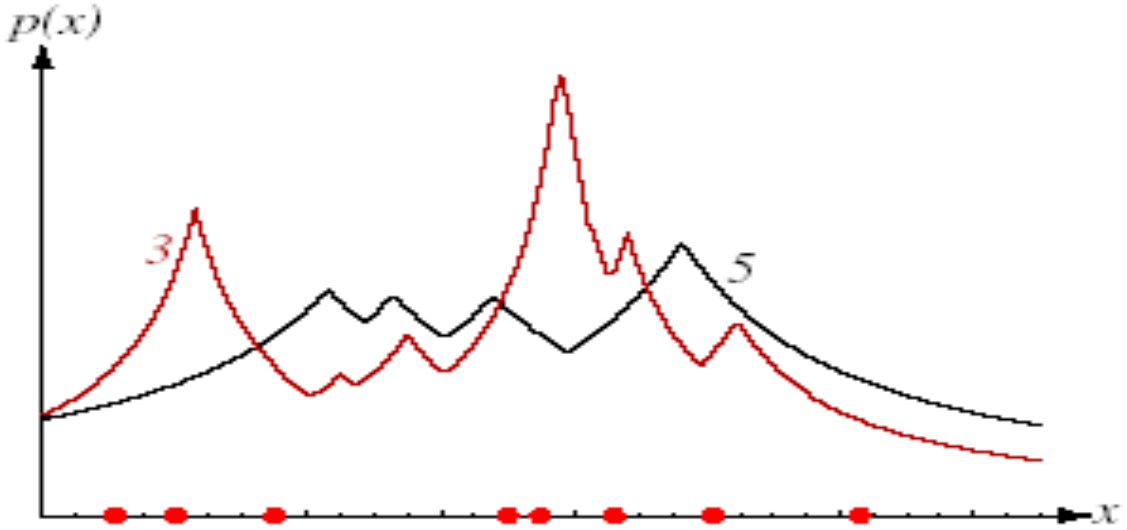
Unsupervised classification

Using the data set $\boldsymbol{D}$ we try to gain some knowledge about $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$ (e.g., we discover that it is made up of two "clusters", i.e., it is the sum of two Gaussian distributions)

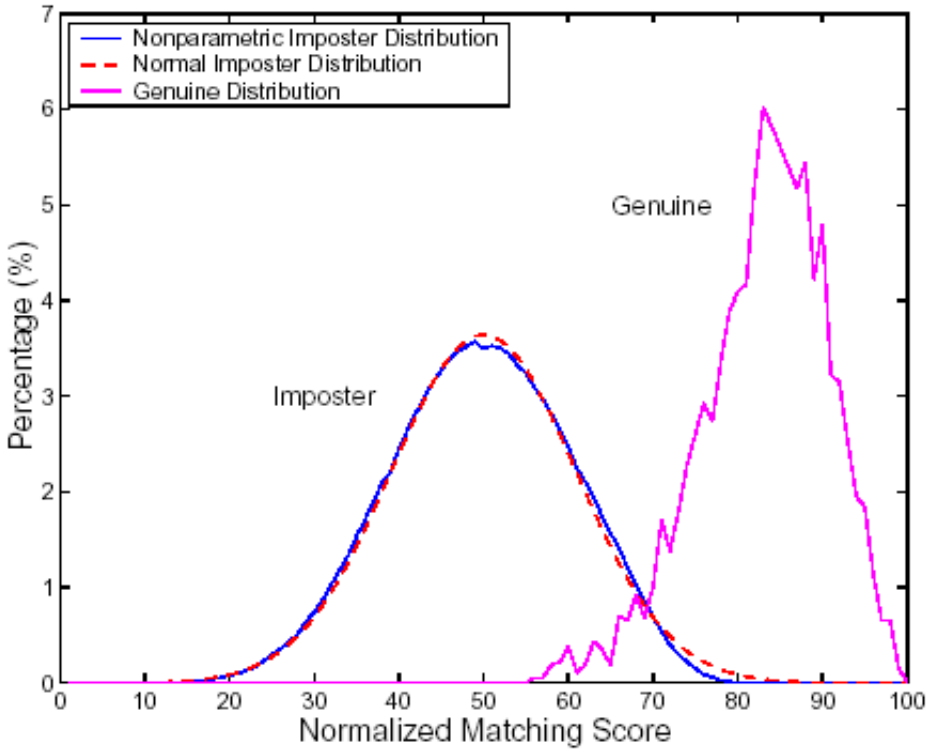
Classification Non-Parametric Techniques

- We know nothing of the distribution $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$, and we are not able to gain knowledge with an unsupervised analysis.
- We use techniques (Part 5) that allow to estimate the densities $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$, or the prosterior probabilities $P(\omega_i/\boldsymbol{x})$, using the data set $\boldsymbol{D}$.

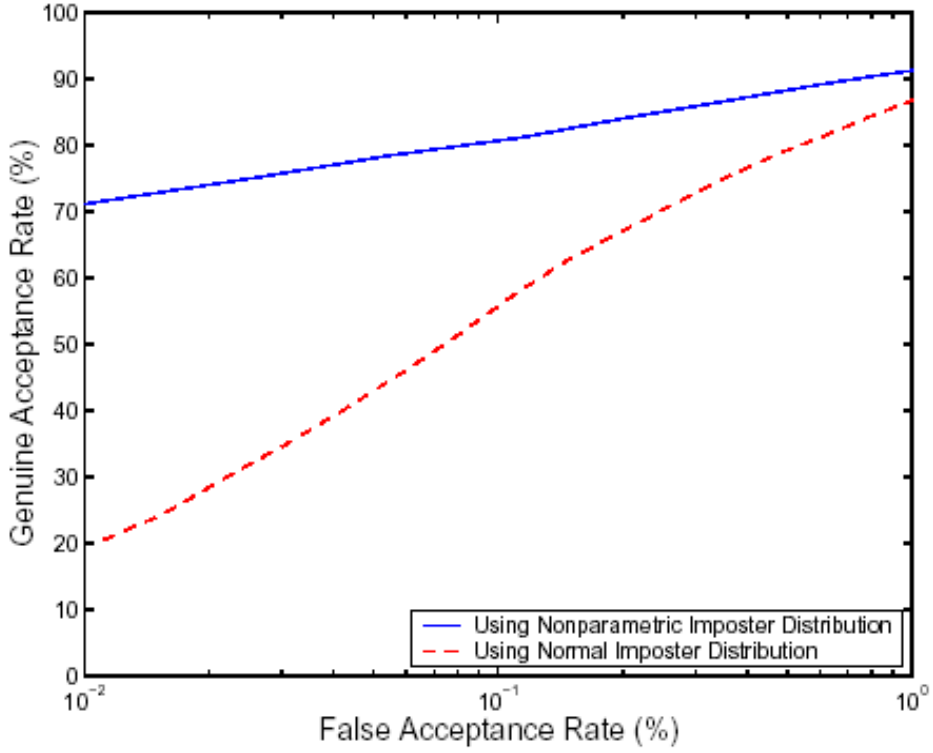
- Non-parametric techniques are aimed to estimate directly the density functions $p(\boldsymbol{x})$



Example of parametric techniques in biometrics



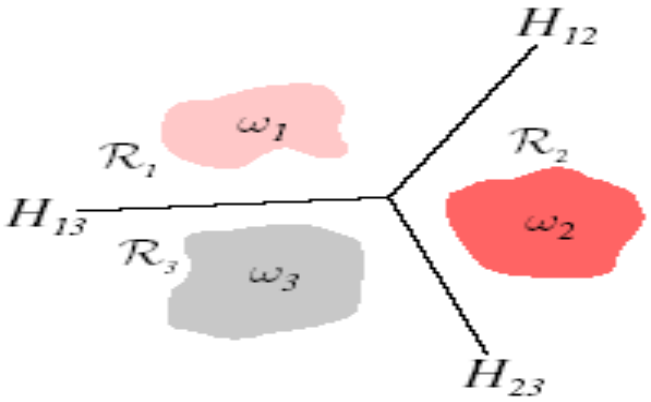
In biometric recognition parametric techniques can be used to model "genuine" and "impostor" distributions



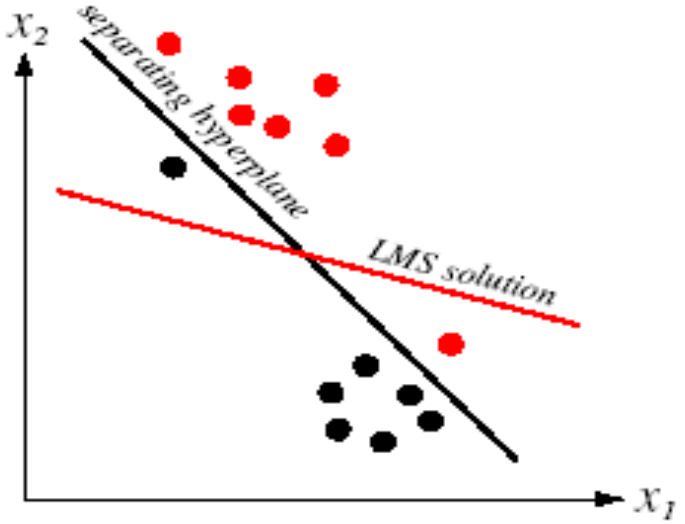
Parametric techniques sometimes provides performances lower than the ones of non parametric techniques

Linear discriminant functions

- In some cases it can be more effective to assume a parametric form of the discriminant functions $g_i(\mathbf{x})$, i=1,..,c, instead of a parametric form of the $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$ (We discuss this in Part 6).
- For example, to assume a linear form of the discriminant functions $g_i(\mathbf{x})$

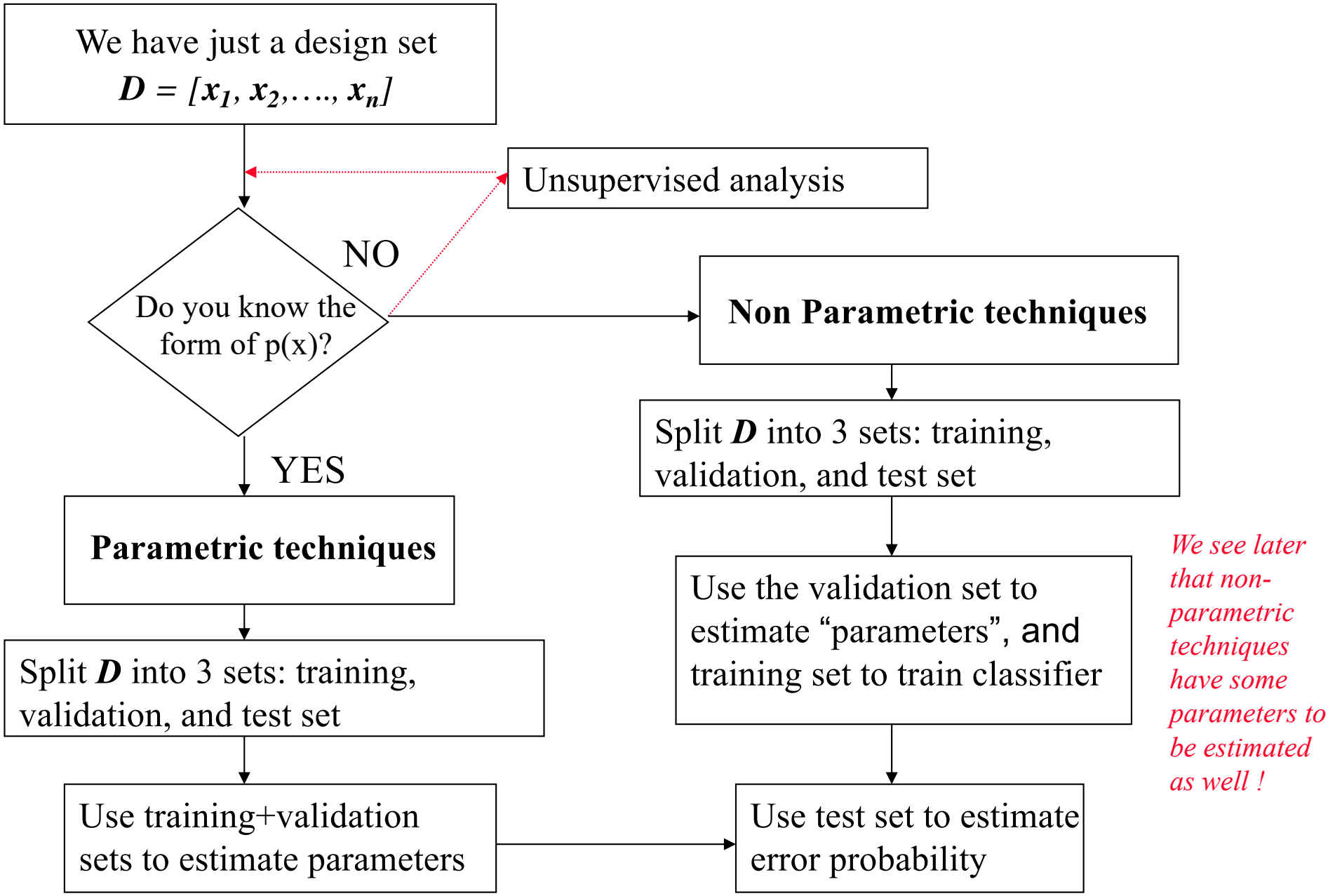


In some cases linear functions allow to disciminate well classes that would be difficult to model by computing the distributions $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$. It is worth noting that, in the end, what we want to do in many cases is just to **"classify", not modelling the $p(\boldsymbol{x}/\boldsymbol{\omega}_i)$!**



Even if a linear discriminat function does not provide the optimal solutions", however the error rate can be acceptable for the task at hand!

Design of classifier: basic design cycle



We see later that non-parametric techniques have some parameters to be estimated as well !

Some notable concepts: “feature” (re)scaling

➢Features used to characterize “patterns” are usually linked to physical measurements which have different scales.

- Given samples in $\boldsymbol{D}$, feature scales can be very different (e.g, height in meters and weight in kg).
- This is due to non-homogenous physical measurements or the intrinsic scales of different features.

➢Solution: normalization, (re)scaling of “features”.

- The “normalization” operation can be regarded as a function h_j applied to *feature* that takes as input the original feature value x_{ij}, and outputs the rescaled(normalized) feature value $x_{ij}' = h_j(x_{ij})$, with h_j being the normalization function ($j = 1, 2, ..., d$).

Some normalization functions

Given $\boldsymbol{D} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$, $\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{id})$ i=1,..,n, normalization functions h_j widely used are the followings:

- We divide the *feature* x_{ij} by maximum value (over $\boldsymbol{D}$):

$$x'_{ij} = \frac{x_{ij}}{x_{j,\max}} \;,\; x_{j,\max} = \max_{k=1,2,\ldots,n} x_{kj}$$

- Divide by maximum range:

$$x'_{ij} = \frac{x_{ij} - x_{j,\min}}{x_{j,\max} - x_{j,\min}} \in [0,1] \;, \begin{cases} x_{j,\max} = \max\limits_{k=1,2,\ldots,n} x_{kj} \\ x_{j,\min} = \min\limits_{k=1,2,\ldots,n} x_{kj} \end{cases}$$

- Divide by standard deviation of feature x_{ij}:

$$x'_{ij} = \frac{x_{ij} - m_j}{\sigma_j} \;, \begin{cases} m_j = \underset{k}{E}\{x_{kj}\} \\ \sigma_j^2 = \underset{k}{E}\{(x_{kj} - m_j)^2\} \end{cases} \quad \begin{cases} \hat{m}_j = \dfrac{1}{N}\sum\limits_{k=1}^{n} x_{kj} \\ \hat{\sigma}_j^2 = \dfrac{1}{N-1}\sum\limits_{k=1}^{n} (x_{kj} - \hat{m}_j)^2 \end{cases}$$

Remarks on normalization

The third normalization method (division by standard deviation) is useful, for example, when *feature* distribution is Gaussian.

- If *feature* x_{ij} has a Gaussian distribution, the normalized *feature* x_{ij}' has a normalized Gaussian distribution.

Normalization must be done using all the patterns available in D and for each *feature* separately.

Hereafter, we assume that all the features used have been properly normalized, and therefore we omit the apex in x_{ij}'.

Some notable concepts: Separation of classes

Definition of separated class:

- In a bi-dimensional *feature* space (d=2), a class is called "separated" if a curve (closed or open) exits so that all the samples of that class lies on the same side of the curve.
- In a d-dimensional *feature* space we have hyper-curves.

Two separated classes can be:

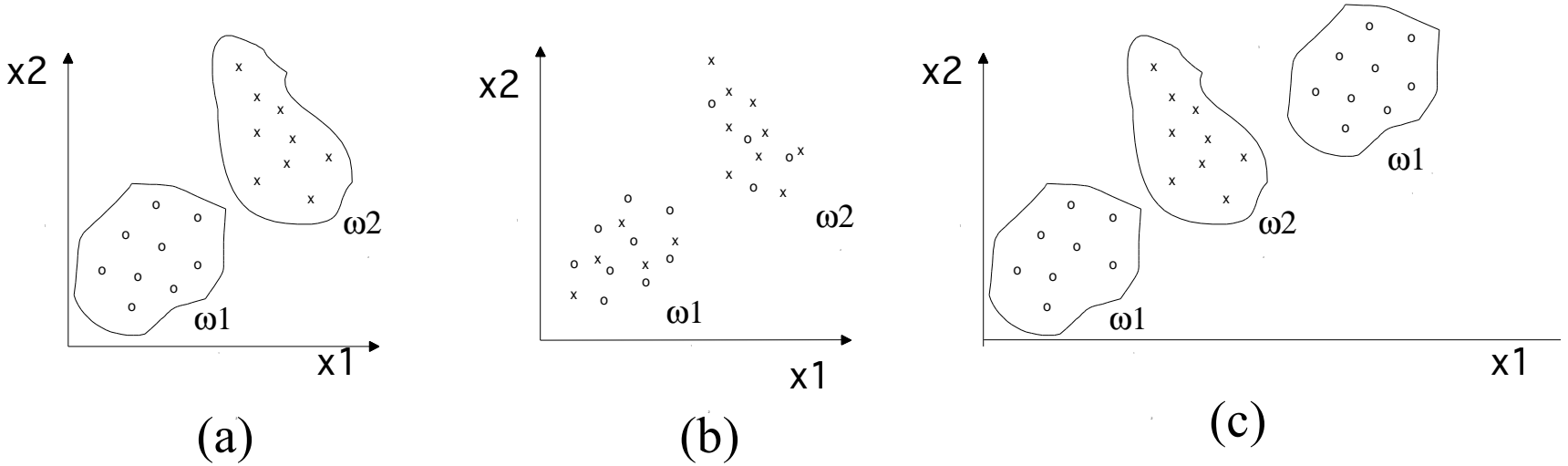
- Linearly separable, if the curve that separates the two classes is a linear function (for d = 2, the curve is a straight line);
- Non linearly separable, the separation needs non-linear curves.

Note that the separation demands that two patterns belonging to different classes do not have the same feature values! So we are speaking of deterministic separation!

Notable concepts: Multi-modal classes

- A data class is multimodal if it contains "clusters" of patterns which are linearly separable or it has different peaks of the density function.

Example

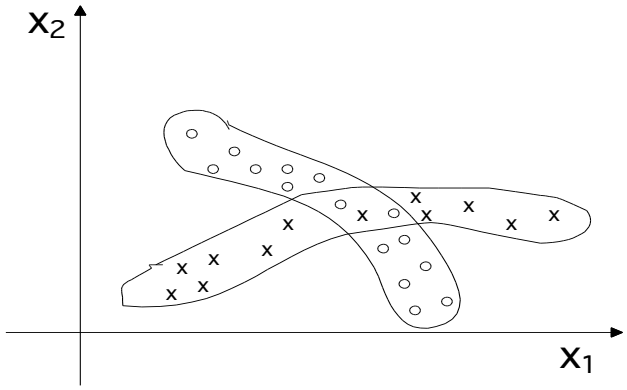


- (a) two linearly separable classes, (b) e (c) two classes non linearly separable. The class ω_1 in (c) is bimodal.
- In (a) and (c) statistical methods work well, the case (b) is much more difficult.

A notable concept: geometrical complexity of classes

Characteristics of a class also dipends on the "geometrical" features of the data distribution in the *feature space*.

- In particular, if classes have elongated distributions and/or are much overlapped, some techniques work poorly.
- Example — it is difficult to discriminate samples in regions where the two classes are very overlapped.



- Each class in the figure have a privileged direction in the *feature space*. *Features* have a very high correlation (conditional correlation given the class).

Correlation Coefficient

Correlation between two *features* x_i ed x_j can be measured by the coefficient of correlation ρ_{ij} $(i, j = 1, 2, ..., d)$.

- It is linked to the variance

$$\sigma_{ij} = E\{(x_i - \bar{x}_i)(x_j - \bar{x}_j)\}$$

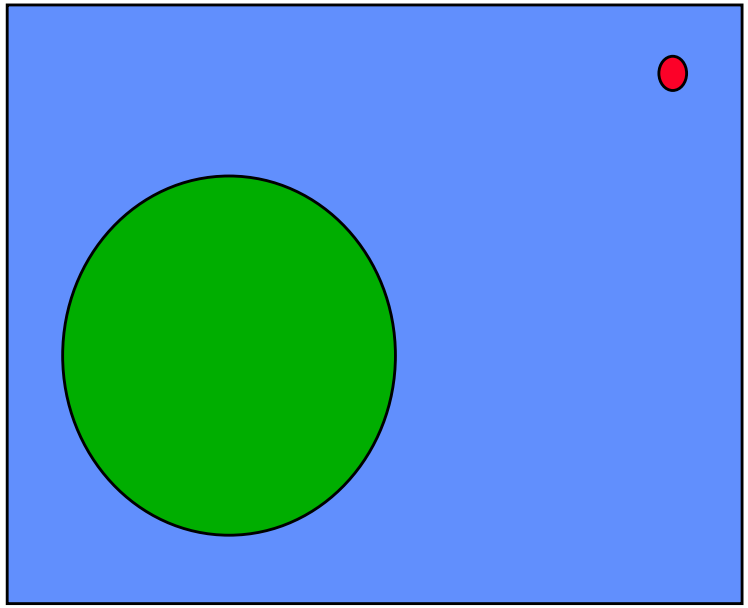
and the feature variances σ_{ii} and σ_{jj} by:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{jj}\sigma_{ii}}}$$

- If d is the *feature number*, $[\rho_{ij}]$ is a squared matrix $d \times d$, con $|\rho_{ij}| \leq 1$ $i, j = 1, 2, ..., d$ e $\rho_{ii} = 1$ (main diagonal) $i = 1, 2, ..., d$.
- *feature* x_i and x_j are correlated if $|\rho_{ij}|$ has a high value (e.g. > 0.8).

The analysis of correlation can be done for each class and for the whole data set.

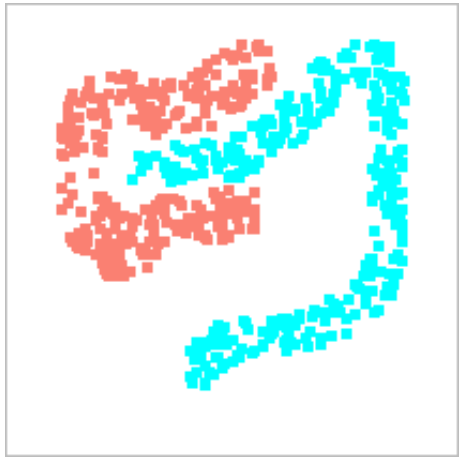
Notable concepts: Geometrical vs. Probabilistic complexity

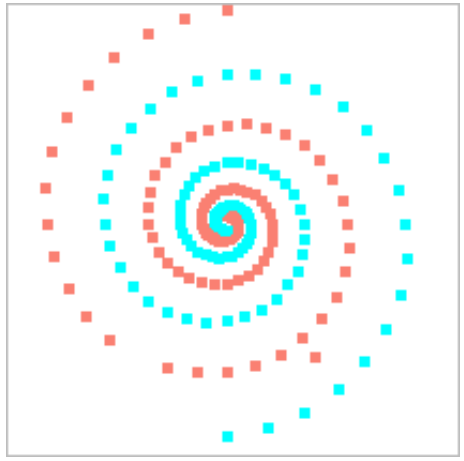


Probabilistic complexity

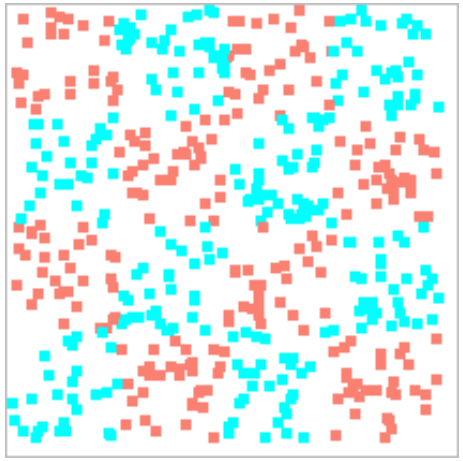
I must recognize one pattern out of one million! Two very unbalanced classes!

The problem has simple geometrical features, but it is very hard!

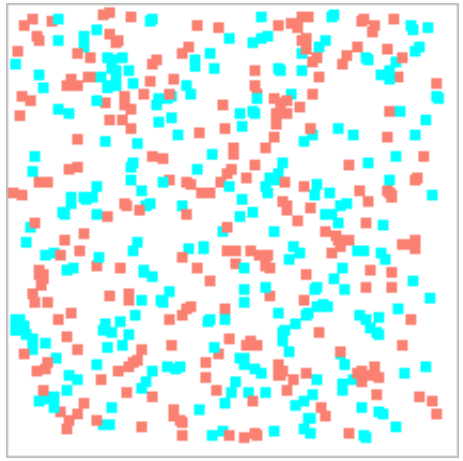




Square 4x4



Square 10x10



Example of **Geometrical Complexity**